What is claimed is:

1. A method comprising:

providing a test hierarchy, the test hierarchy including a collection of test methods, each test method calling test assertion methods for checking correctness of production code, each test assertion call defining a test execution flow within the test hierarchy in an event of a failure; and

controlling the test execution flow inside the test hierarchy according to a parameter of the test assertion method call in the event of the failure.

2. The method of claim 1 wherein the parameter causes instructions to continue or to abort at any level of the test hierarchy after the event.

3. The method of claim 1 wherein a test assertion method includes instructions for verifying an expected state of the production code.

4. The method of claim 1 wherein the test hierarchy is implemented using an object oriented programming language including at least one of ABAP (advanced business application program language), Java programming language, C++ programming language, and C# programming language.

5. The method of claim 1 wherein the test hierarchy is implemented using a procedural programming language including at least one of C programming language, Fortran programming language and Pascal programming language.

6. The method of claim 1 wherein the test hierarchy is implemented in a unit test environment.

7. A computer program product, tangibly embodied in an information carrier, the computer program product comprising instructions operable to:

provide a test hierarchy, the test hierarchy including a collection of test methods, each test method calling test assertion methods for checking the correctness of production code,

each test assertion call defining a test execution flow inside the test hierarchy in an event of failure detection; and

control the test execution flow within the test hierarchy according to a parameter of the test assertion method call in response to detecting the event.

8. The product of claim 7 wherein the parameter causes the test execution flow to continue or to abort at any level of the test hierarchy after the occurrence of the event.

9. The product of claim 7 wherein a test assertion method includes instructions for verifying an expected state of the production code.

10. The product of claim 7 wherein the test hierarchy is implemented using an object oriented programming language including at least one of ABAP (advanced business application program language), Java programming language, C++ programming language, and C# programming language.

11. The product of claim 7 wherein the test hierarchy is implemented using a procedural programming language including at least one of C programming language, Fortran programming language and Pascal programming language.

12. The product of claim 7 wherein the test hierarchy is implemented in a unit test environment.

13. A computer system comprising:

a test hierarchy, the test hierarchy including a collection of test methods, each test method calling test assertion methods for checking the correctness of production code, each test assertion call defining the test execution flow within the test hierarchy in an event of a failure; and

a means for controlling a flow inside the test hierarchy according to a parameter of the test assertion method call in response to the event.

14. The system of claim 13 wherein the test assertion is called with a parameter causing the test execution to continue or to abort at any level of the test hierarchy after the event.

15. The system of claim 13 wherein a test assertion method includes instructions for verifying an expected state of the production code.

16. The system of claim 13 wherein the test hierarchy is implemented using an object oriented programming language including at least one of ABAP (advanced business application program language), Java programming language, C++ programming language, and C# programming language.

17. The system of claim 13 wherein the test hierarchy is implemented using a procedural programming language including at least one of C programming language, Fortran programming language and Pascal programming language.

18. The system of claim 13 wherein the test hierarchy is implemented in a unit test environment.